Setup Docsify with Podman and Integrate with Github

Linux Distribution

- OS Name : Ubuntu 20.04.6 LTS

System Configuration

- RAM : 5.6 GiB
- CPU : 12
- STORAGE : 512.1 GB

Prerequisites tools

- Podman
- Github

Docsify

Docsify a popular documentation site generator. Docsify allows you to easily create documentation websites from simple Markdown files.

Docify is a completely flexible tool that automates the preparation of recurring documents.

To use Docsify in Ubuntu, you'll need to follow these steps:

Podman

Podman is an open-source container management tool that provides a way to manage containers on Linux systems.

GitHub

GitHub is a platform and cloud-based service for software development and version control using Git, allowing developers to store and manage their code.

Step 1:- Install Podman:

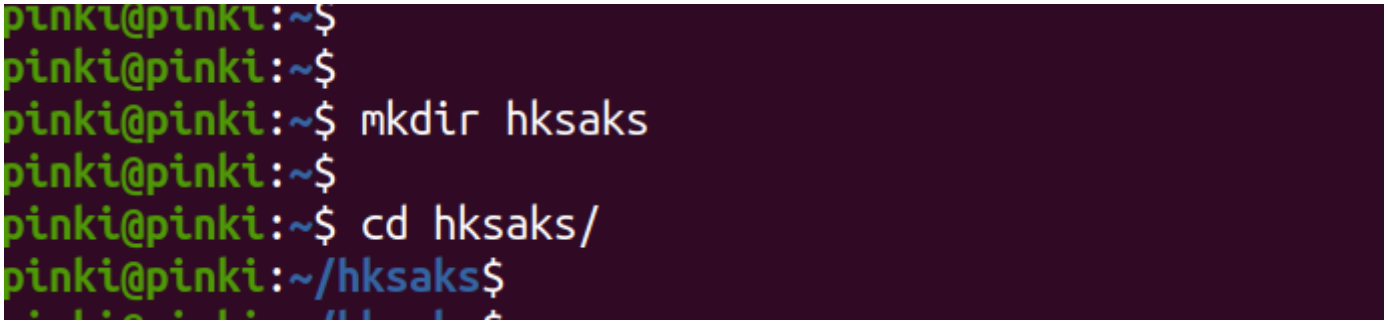
```
sudo apt -y install podman
```

command is used to install Podman, on a Linux system.

Step 2:- Create a documentation directory:

```
mkdir hksaks
```

We use "mkdir" command for creating the directory.



Step 3:-Create Dockerfile:

```
vim Dockerfile
```

The content you've provided is a Dockerfile. A Dockerfile is a script that contains a set of instructions for building a Docker image. Each line in a Dockerfile represents a step in the image-building process.

- **FROM node:latest:** It uses the official Node.js image with the "latest" tag, which means it will use the latest available version of Node.js as the base for your image.

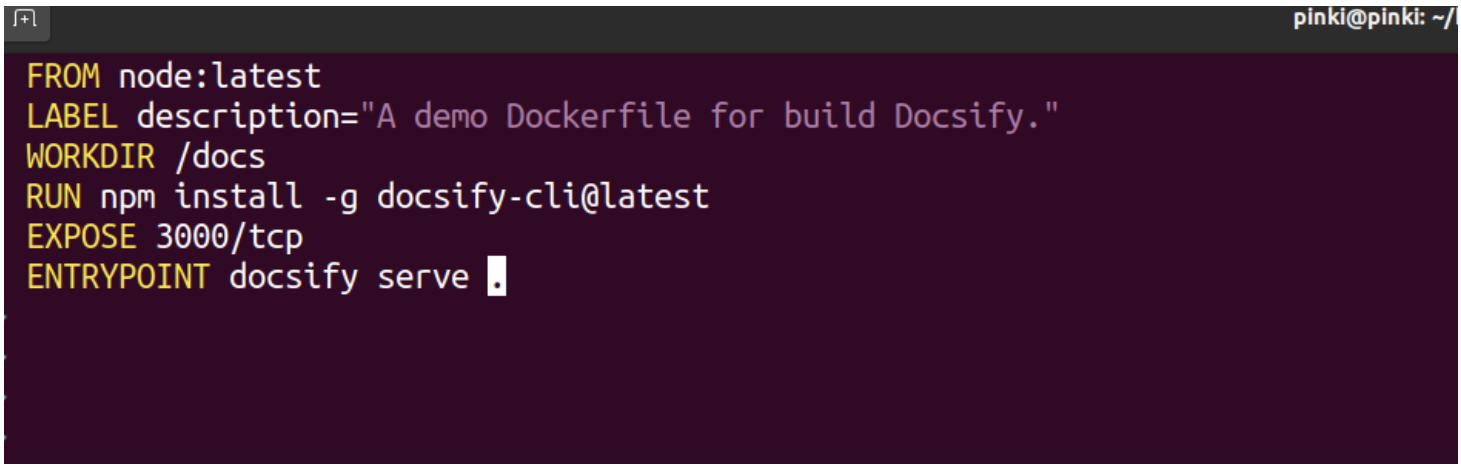
- **LABEL description="A demo Dockerfile for building Docsify.":** This line adds metadata to your image in the form of a label. Labels are used to provide additional information about the image. In this case, it describes the purpose of the Dockerfile, which is to build Docsify.

- **WORKDIR /docs:** This line sets the working directory inside the container to /docs.

- **RUN npm install -g docsify-cli@latest:** This instruction runs a command inside the container. It uses npm (Node Package Manager) to install the docsify-cli package globally. This package is required to run Docsify..

- **EXPOSE 3000/tcp:** This instruction declares that the container will listen for incoming network connections on port 3000.

- **ENTRYPOINT docsify serve .:** This instruction specifies the command that should be executed when a container is started from this image. It tells the container to run docsify serve ., which starts the Docsify server, serving documentation from the current directory (.).



Step 4:-Create file:

```
touch index.html
```

```
touch README.md
```

By using this command create file in directory.

```
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$ touch index.html
pinki@pinki:~/hksaks$ touch README.md
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$ ls
Dockerfile  index.html  README.md
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$
```

Step 5:-Build docker image:

```
docker build -f Dockerfile -t docsify/demo .
```

The command you provided, docker build -f Dockerfile -t docsify/demo ., is used to build a Docker image from a Dockerfile.

```
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$ sudo docker build -f Dockerfile -t docsify/demo .
Sending build context to Docker daemon  3.072kB
Step 1/6 : FROM node:latest
 ---> 341640cdfda9
Step 2/6 : LABEL description="A demo Dockerfile for build Docsify."
 ---> Using cache
 ---> c71fa7526f91
Step 3/6 : WORKDIR /docs
 ---> Using cache
 ---> ebafaac26be1
Step 4/6 : RUN npm install -g docsify-cli@latest
 ---> Using cache
 ---> 962159d309c1
Step 5/6 : EXPOSE 3000/tcp
 ---> Using cache
 ---> a8bdb614d2ca
Step 6/6 : ENTRYPOINT docsify serve .
 ---> Using cache
 ---> ea2a23f9bb46
Successfully built ea2a23f9bb46
Successfully tagged docsify/demo:latest
```

```
podman images
```

This command provides information about the images you have downloaded or built using Podman, including details such as the repository, tag, image ID, and size.

```
pinki@pinki:~/hksaks$ podman images
REPOSITORY                  TAG         IMAGE ID      CREATED        SIZE
localhost/docsify/demo      latest      1a295879ea34  39 hours ago   1.15 GB
localhost/my-docsify-image  latest      41b3b0e49d32  43 hours ago   943 MB
docker.io/library/node      latest      341640cdfda9  4 days ago     1.12 GB
docker.io/library/node      14          1d12470fa662  4 months ago   937 MB
```

Step 6:-Podman run :

```
podman run -itp 3000:3000 --name=docsifya -v /home/pinki/hksaks:/docs docsify/demo
```

The podman run command is used to run containers in Podman

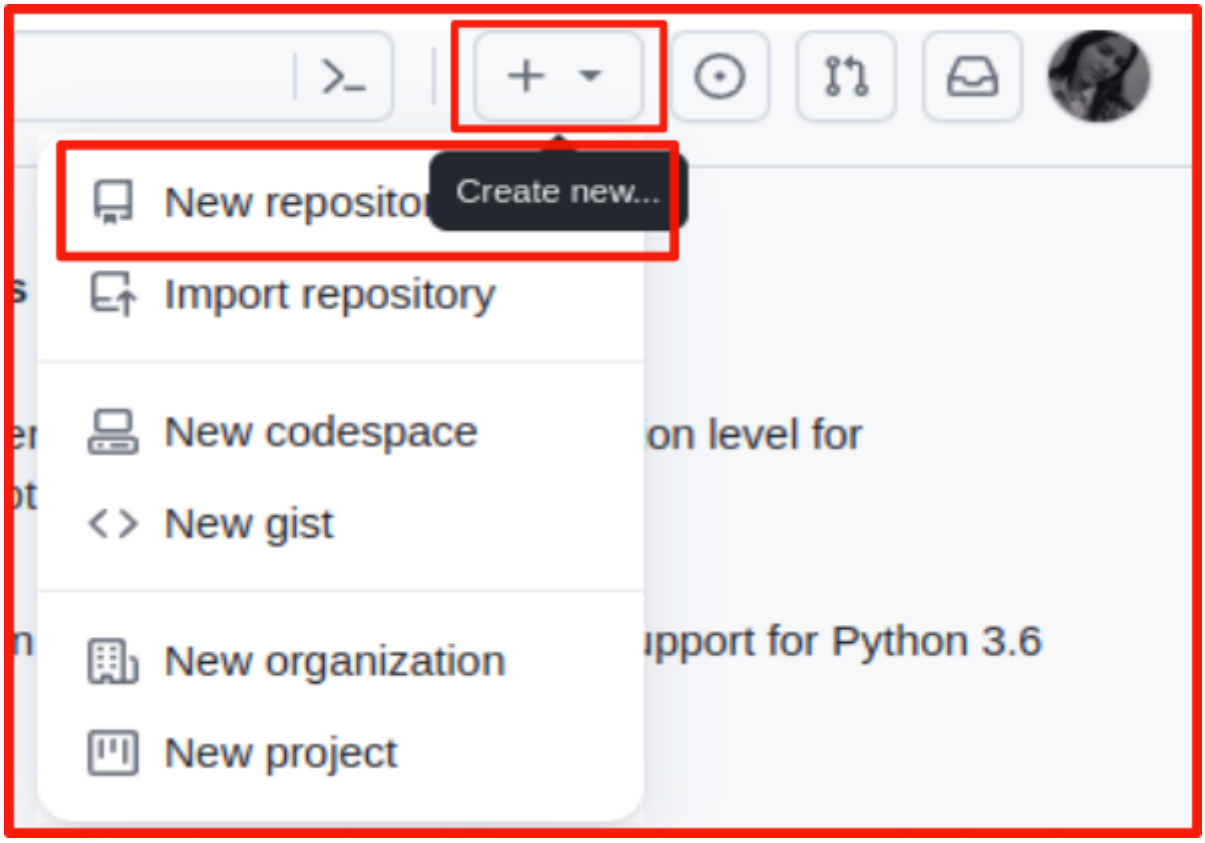
```
pinki@pinki:~/hksaks$ podman ps -a
CONTAINER ID  IMAGE                              COMMAND              CREATED         STATUS                     PORTS                   NAMES
48ffcf241fba  localhost/my-docsify-image:latest  docsify serve ./d...  5 hours ago     Exited (1) 5 hours ago                             docsify
158b4415328d  localhost/docsify/demo:latest                           9 seconds ago   Created                    0.0.0.0:3000->3000/tcp  docsifya
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$
```

Step 7:-Output of the Preview :

Run docsify on localhost

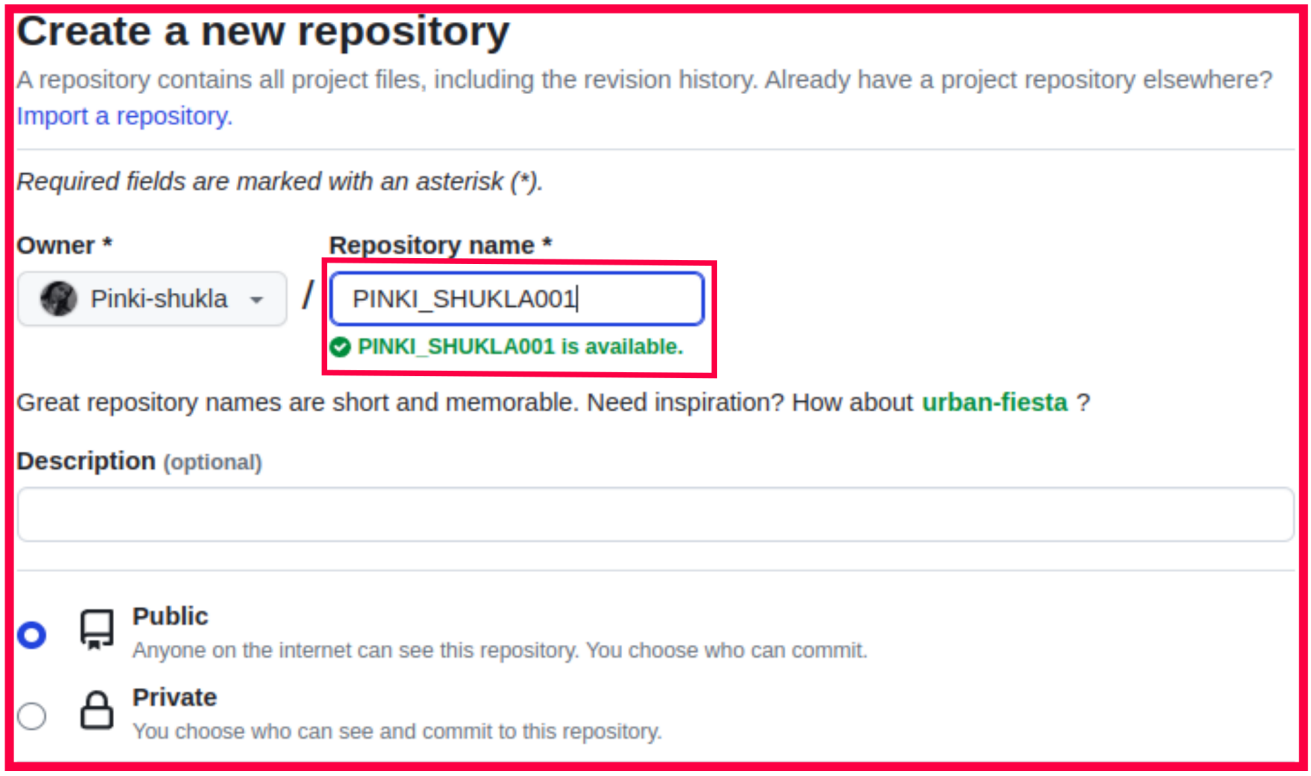
Github

Step 1:-For making a new repository, click on "+" icon:



)

Step 2:-Enter your repository name and give permission to the public.



Step 3:- Clone the git repository

```
git clone https://github.com/username/repository.git
```

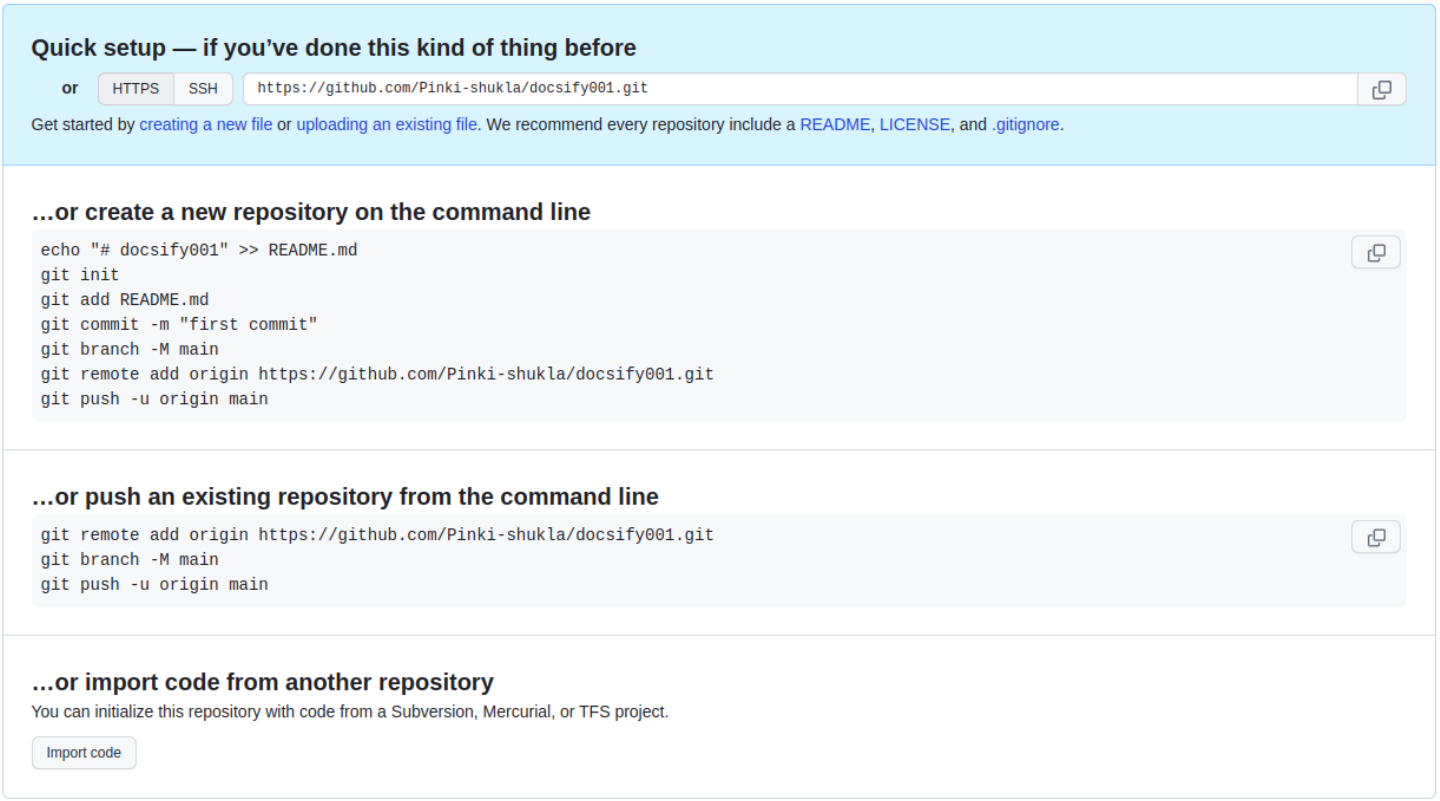
The git clone command is used in Git version control to create a copy of a remote Git repository on your local machine.

Replace username with the owner's GitHub username and repository with the name of the repository. You can obtain the URL from the repository's GitHub page or another Git hosting service.

```
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$ sudo git clone https://github.com/Pinki-shukla/ak.git
Cloning into 'ak'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 591 bytes | 591.00 KiB/s, done.
pinki@pinki:~/hksaks$
pinki@pinki:~/hksaks$
```

Step-4 In this step, you need to follow the command that is provided by the github repository:

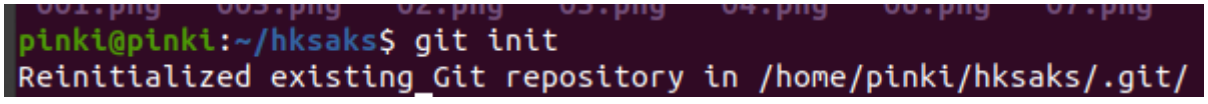
Follow each commands to copy and paste.



The git init command is used to initialize a new Git repository in a directory on your local machine. When you run git init in a directory, it sets up the necessary files and directories that Git uses to manage version control for your project.

Git Repository: A Git repository is a version control system that allows you to track changes in your files and collaborate with others on software development projects. It stores a history of changes made to files in your project, making it easy to manage and track revisions.

```
git init
```



When you make changes to your files, such as editing or adding new content to README.md, you need to use git add to inform Git that you want to include these changes in the next commit

```
git add README.md
```

The command git commit -m "first commit" is used to save or record changes to the repository, describing the changes you've made.

```
git commit -m "first commit"
```

The git branch -M main command is used to rename the default branch of your Git repository. This command is commonly used to change the name of the default branch from master to main.

```
git branch -M main
```

The git remote add origin command is used to connect your local Git repository with a remote repository on a platform like GitHub. The origin in this command is typically a short name used to refer to the remote repository.

```
git remote add origin https://github.com/Pinki-shukla/docsify001.git
```

The git push -u origin main command is used to push the local commits in your repository's main branch to the remote repository specified by the origin remote. The -u flag is used to set up tracking between the local main branch and the remote main branch, making it easier to push and pull changes in the future.

```
git push -u origin main
```